Creating an Environment for Linking Knowledge-based Systems to a Clinical Database: A Suite of Tools

Adam Wilcox, M.A., George Hripcsak, M.D., Cynthia Chen, M.S.
Department of Medical Informatics, Columbia University, New York, NY

A difficulty in using knowledge-based systems has been linking them to clinical databases. The challenge is in making a correct mapping from the data in the knowledge base to the data in the database. At Columbia-Presbyterian Medical Center, we have built a suite of tools developed to create queries that address this challenge. The tools were designed to allow users to easily retrieve data from the database without requiring the users have extensive database and vocabulary knowledge. The tools help users write correct queries (Query Builder), find correct terms in the clinical database (MED Browser), aggregate the resulting data into a useful form (Clinical Database Browser), and allow the user to test the query within the environment of the knowledge-based system (Event Playback). The tools have been in use for one year.

INTRODUCTION

Despite the multitude of systems that have been reported in the literature over 30 years, and despite the number of evaluations showing that knowledge-based systems (KBSs) ought to be useful [1-4], a tour of the average hospital would convince one that these systems have little use in clinical care. Basic KBS research may reduce this discrepancy, but it will not eliminate it. Part of the problem lies not within the KBS itself, but in the link between the KBS and the clinical environment [5,6]. The routine availability of more and more clinical data in coded electronic form is reducing one of the hurdles to using KBSs: the manual entry (and re-entry) of clinical data. For example, the creators of QMR reported that one of the factors discouraging the use of QMR is the manual entry of data [7]. To reap this benefit of availability, however, a major effort is required to map conceptual entities in the KBS to actual entries in the clinical database.

At Columbia-Presbyterian Medical Center (CPMC), ten years have been invested in building a clinical information system. Automated decision support for this system was developed using the Arden Syntax for Medical Logic Modules [8,9], which is being used at several institutions for alerts, interpretations, diagnosis scoring, protocols, and clinical research. One fact that has been apparent through this development is that the largest challenge to effective decision support is getting the data. Even when the data are available in the clinical database, finding where those data are stored and converting those data into a form acceptable to the KBS requires extensive knowledge of the database and the vocabulary. A current example of a KBS with such a problem is the CPMC Medical Logic Module (MLM) knowledge base. The CPMC experience with this KBS has been that the writing and testing of queries consumes more time than all the other MLM tasks combined [10]. The results of a knowledge sharing study indicate that differences in vocabularies cause the greatest number of modifications necessary to share MLMs, and that differences in database organization cause the largest single modifications [11]. Therefore the focus of this work has been to build better database retrieval tools, rather than building better knowledge base tools. The group of tools built have two specific aims to help database retrieval. First, Query Builder and the MED Browser work to increase the accuracy while reducing the writing time and technical skills required to author clinical database queries. Second, the Clinical Database Browser and Event Playback facilitate the testing of queries, and improve the match between a query's result and the needs of a KBS. All of the tools are applied to CPMC's MLM knowledge base.

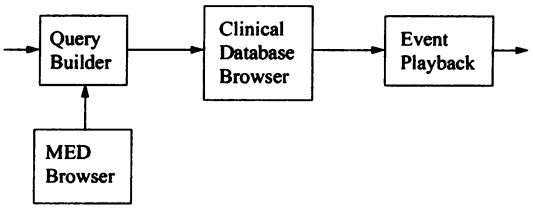

Figure 1. The tools help the user during the process of linking the KBS to the clinical database. Query Builder first helps create a correct query, and the MED Browser helps find correct database terms while building this query. The Clinical Database Browser then helps the user test the result of the query, and Event Playback then can test the query within the environment of the KBS.

1. QUERY BUILDER

In order to link a KBS entity to a clinical database entry, the query to the database must first be defined. This involves specifying the desired KBS entity, reviewing the entries available in the clinical database, choosing the appropriate terms and retrieval methods, and assembling a syntactically correct

1091—8280/97/$5.00

query. This process requires knowledge of the organization of the database, the terminology used to store the information, and the syntax of the query.

Query Builder is a web-based application, written in Perl, that allows a less experienced author to write a query. It requires less knowledge of the database organization, uses a familiar interface, and assembles a valid query itself. With the application, an author moves through sections of an HTML form that ask specific questions about the data the author wants to retrieve. The author is not required to know the details of the database design, the underlying vocabulary or the exact syntax of the query language.

For example, if an author were trying to create a query for the results of serum and plasma creatinine tests, she would first choose the Laboratory module of Query Builder. (Other modules include Pharmacy, MLM Messages, and Demographics.) She is asked to either choose the desired test from a list of common tests, or enter the medcodes of the desired tests (see section 2 below). Accompanying selections allow her to constrain the query to specific batteries of tests, if desired. Then she can select the information to be retrieved about the selected data, as well as time constraints for the data. The remaining sections of the form allow the author to specify the number of instances returned and sorting order. She can then submit these data, and a query is returned as useable code. Each case has a default selection, which was determined by what we thought would be the most requested query attributes.

With this information, Query Builder can create the correct syntax for the specified query. The author may choose one of two types of output, depending on how the query will be used. First, Query Builder can produce an Arden Syntax query, which can be copied from the browser display and inserted directly into the MLM's data slot. The body of this query is defined in a construct called the curly bracket expression [8]. This expression is CPMC-specific, a situation that is unavoidable until there is better agreement among vocabularies and among databases. Otherwise, Query Builder will produce a properly formatted HL7 query, which can be inserted into an application program. HL7 is intended for communication between computers rather than reading by human beings, so its queries are not amenable to modification. The result of either output is a correctly formatted query, containing the correct vocabulary elements and data access information, that will access the specific clinical data intended by the author. The author is not required to completely understand the CPMC clinical data model to query the data.

The modules of Query Builder are tailored to the type of query the user is creating. Pharmacy data contain different information than Laboratory or Demographics, and the modules reflect that difference. However, the appearances and user interfaces of the modules are similar.

2. MED BROWSER

The query returned by the Query Builder is designed to retrieve data in electronic form, and thus requires that the correct codes and definitions of clinical terms be used when building the query. CPMC's medical vocabulary is represented by the Medical Entities Dictionary (MED) [10,11]. It defines all terms stored in the clinical database, it maps database terms to the terms used in ancillary departments, it provides mechanisms for users to find terms, and it provides tools for maintaining the vocabulary. The structure of the MED is a semantic network of medical terms that are classified hierarchically. For example, the medical terms *Whole Blood Count* and *Prothrombin Time* are nodes in the network that are descendants of the term *Laboratory Diagnostic Procedure*. As a descendant of *Laboratory Diagnostic Procedure*, *Whole Blood Count* inherits slots such as *Has Parts* and *Specimen*, which are then used to establish semantic relationships to other medical terms (*Hemoglobin* and *Blood*, respectively). The MED semantic network is built upon the Unified Medical Language System (UMLS) Semantic Network of the National Library of Medicine [11,12].

The MED Browser allows developers and users to find the terms that they need to retrieve and store clinical data. It is also a web-based application, and is linked to the Query Builder. The Browser uses HTML tables to display the "is_a" relationship between terms. In the left column are the semantic parents of the selected term. The center column contains the term and its siblings (terms that are children of any of its parents), and the right column contains the children of the term. A user can get to other terms by traversing the network, done by selecting any of the displayed terms, which creates a table focused on that term. While other graphical browsers of the MED exist with only slight differences in display and function [13], the MED Browser offers a special search algorithm.

Searching Algorithm

Any of the set of browsers use lexical matching and synonymy to find candidate terms in the network. The author can decide whether these terms are correct by looking at the properties of each term. However,

this can be tedious when the list of terms is large. The MED Browser assists the user in searching for a specific term by using the semantic network of the MED itself to find the "most general" term.

First, a list of terms is obtained using lexical matching. These terms are initially ordered by increasing medcode. This initial ordering is not useful, since there is little correlation between the probability that a specific term is desired by the user (according to the search string entered) and the medcode of the term. To accomplish a better ordering, the descendants of each matched term are searched. Each term is scored according to the number of descendants it has which are also in the list of original matched terms. The terms with the higher score are "more general" according to the terms listed. The terms are presented in order of decreasing score. In the case that two terms have the same score, they are ordered by medcode. The keyword may exactly match the name of one of the medical entities. In this case the program assumes this matching term is the correct one, so that term becomes the highest scoring term listed. The rest of the terms are ordered by decreasing matched descendants, as above. In either case, the table is focused on the highest scoring term. The list of matching terms is also presented, and the user can pick other terms if she is not satisfied with the first term presented.

3. CLINICAL DATABASE BROWSER

In order to test the queries and improve the match between a query's result and the needs of a KBS, the data that are returned must be characterized. If the data can be characterized, the author can more easily review the data to determine whether they match the expectations of the KBS. The purpose of the Clinical Database Browser (CDB) is to present the result in a useful form, by using the MED to aggregate complex nominal data.

The aggregation of numeric data is familiar and has been implemented in many medical research systems [14-16]. Given a population of patients, a query for a numeric attribute like the serum potassium will return a long list of numbers that is difficult to grasp or use. Through aggregation, one can determine the collective properties of the population. The result is a manageable summary that is useful for assessing the result of an intervention, comparing populations, setting thresholds for MLM parameters, and so on. Medical research systems include statistical aggregation (mean, standard deviation, regression, ...). Non-numeric data that have an intrinsic order can be analyzed via non-parametric techniques. Graphical display of data, such as scatter plots and histograms, is often the most efficient way to express the collective properties of the population.

Like numeric data, nominal data can be overwhelming. Unlike numeric data, nominal data often lack the implicit ordering among values and the natural distance metric that make numeric data easy to aggregate. Some nominal attributes, like gender and marital status, have few categories, and aggregation is simple; a histogram usually suffices. In medicine there are many important nominal attributes, like diagnosis and physical finding, that have numerous categories. In these instances, a histogram fails to reduce the amount of data enough to be comprehensible. For example, a simple histogram that shows how many times each unique discharge diagnosis appears in a patient population may have hundreds of bars (such a query is quite common when writing MLMs). A solution is to lump individual codes into logical groups to cut down on the number of bars.

The CDB accepts a list of codes, and uses the MED to group these codes automatically. All codes that can be stored in the clinical database are represented as terms in the MED, and a great deal of work has been done to classify these terms logically. The CDB uses the MED to display the relationship among the terms while at the same time displaying how many patients in the target subpopulation are related to each term. In effect, the MED provides a partial ordering and distance metric for nominal data. A MED term becomes a categorical class, and all the descendants of the term are members of that class.

The CDB (see Figure 2) looks very much like the MED Browser (indeed, the two were developed simultaneously, and the search algorithm of the MED Browser is included in the CDB). The differences are that 1) the CDB only displays terms relevant to the data displayed, 2) the CDB displays four columns instead of two (parents, term and siblings, children, grandchildren), and 3) the CDB displays two histogram bars beneath each term. The lower bar indicates how many results are contained in the class or any of its descendants, while the upper bar indicates how many results are represented by that term alone, not including descendants. Next to each is a number representing either the exact number of terms represented, or the percentage (depending on the user's preference). The graphs can use either a linear scale or a logarithmic scale. The user can traverse the MED or search for specific terms, to see the proportions of codes grouped into each class.

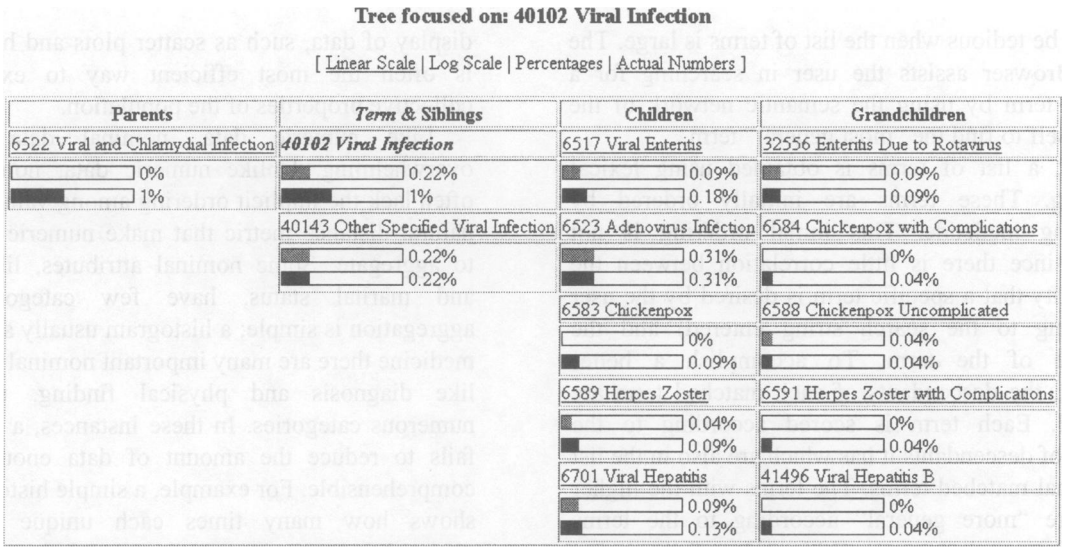

Figure 2. Sample table from CDB. Many individual diagnoses can be lumped together into one category or group. Graphs are shown using a logarithmic scale. (See text.)

4. EVENT PLAYBACK

Another step to testing queries is to test within the environment of the KBS, to see whether the KBS performs as expected. With MLMs, this is generally the most time consuming step. The best way to test an MLM is to turn it on (i.e., let the event monitor run it in real time), but send the generated messages to the MLM author instead of the patients' clinicians. The author can review the messages to see whether they were appropriate. The problem is that most MLMs fire rarely, so this sort of test can take two or more weeks to gather sufficient information. MLMs are refined based on the results of the test, and then they are retested. The entire process can take months.

The Event Playback tool replays medical events as if they were occurring in real time. (Most events are clinical database transactions, but it is also possible to log events such as a user signing onto the system.) An author can choose any time period within the previous year and test the MLM as if it had been running during the original time period. Since only a small number of events are relevant to a given MLM, a two-week period can be replayed in minutes. While it is true that the knowledge engineer was not previously forced to sit at a terminal for the two week test period, the ability to test an MLM in one session is far less disruptive and encourages more complete testing (e.g., gather more information by testing it over a longer time period). For the group that requests an MLM, the two week savings is real. Because one can replay the same time period repeatedly, it is easier to assess how revisions affect the MLM in particular situations. For example, if a test reveals that an MLM behaved incorrectly for a particular patient at a particular time, the revised MLM can be tested under the same circumstances. By trying several versions of an MLM where each version has a different threshold, a receiver operating characteristic curve can be generated for it. Event Playback also helps with the false negative problem. When an MLM generates a message that should not have been generated (false positive), it is easy to review the patient data and discover that the message was inappropriate. But when an MLM fails to generate a message (false negative) the author does not know which patient to look at or even whether such a false negative occurred. Using Event Playback, the author can run a revised MLM that uses relaxed thresholds or independent algorithms in order to point out patients for whom there should have been a message. For example, the reliability of the admit diagnosis for detecting admissions for labor and delivery can be measured by comparing an MLM that reports the admit diagnosis to an MLM that reports the mothers of all newborns in the hospital.

An important component to the Event Monitor is the event log of past events. An upload of data of any type to the clinical database is classified as a clinical event. The logging facility subscribes to the current broadcast system, which allows ancillary departments to receive notification of all clinical events via the HL7 protocol. The event logging program saves only those fields necessary to simulate the broadcasting of events (i.e., time, event type, event object, medical record number, and data key). An uncompressed monthly log requires about 70 Mbytes of storage space, though this can be compressed (using gzip) to ~10 Mbytes.

To run the Event Playback tool, an author enters the test MLMs and picks any period of time up to the present. Event Playback reviews the event log for the specified time, and simulates the events relevant to the MLM. For each event, Event Playback causes the CPMC event monitor to trigger the MLM being tested. When an MLM is triggered in this way, its queries can only retrieve the data that were available at the time when the event originally occurred. Messages generated by the MLM are collected and sent back to the author.

DISCUSSION

The suite of tools developed were designed to help create an environment where authors could retrieve data from a database without requiring extensive database and vocabulary knowledge, and reduce authoring time. This is possible because the tools are driven by the data model and vocabulary of the CPMC clinical information system. Authors can view the data as they are modeled and organized by the system, and can conform their data model to the clinical database model. Such an environment can reduce misconceptions of data meaning and erroneous queries. The tools have been running in production mode for one year, and have been used for MLM development and other purposes. Query Builder can create HL7 queries that may be used in applications other than MLMs. The MED Browser is being used to browse the MED structure for other purposes, especially by new students who are trying to conceptualize the MED. The Event Playback logs have been used to track the operation of the clinical information system.

Though the tools successfully address the problem of linking between the KBS and clinical database, there remain important challenges. First, maintenance of such tools is difficult. The tools rely on static components, though they are not always so. For example, a recent upgrade of the CPMC-specific HL7 coding has led to some invalid queries. Unless the tools change to compensate for changes in the underlying components, the tools will eventually be useless. Second, the difficulty of the problem continues to be a challenge. Though the tools are helpful, no one has reached complete success in matching a user's model of data to the database model. The tools can only make it easier.

Acknowledgment

This work was supported by National Library of Medicine Grant R29 LM05627 "Linking Knowledge-Based Systems to Clinical Databases".

References

1. Pestotnik SL, Evans RS, Burke JP, Gardner RM, Class DC. Therapeutic antibiotic monitoring: surveillance using computerized expert system. Am J Med 1990;88:43-8.
2. McDonald CJ, Hui SL, Smith DM, Tierney WM, Coh SJ, Weinberger M, McCabe GP. Reminders to physicia from an introspective computer medical record. Ann Inte Med 1984;100:130-8.
3. Barnett GO, Winickoff RN, Morgan MM, Zielstorff R A computer-based monitoring system for follow-up elevated blood pressure. Med Care 1983;21:400-9.
4. Rind DM, Safran C, Phillips RS, Slack WV, Calki DR, Delbanco TL, Bleich HL. The effect of compute based reminders on the management of hospitaliz patients with worsening renal function. In: Clayton PD, e Proc SCAMC 15, 1992; 28-32.
5. Shortliffe EH. Computer programs to support clinic decision making. JAMA 1987;258(1):61-6.
6. Wyatt J. Computer-based knowledge systems. Lanc 1991;338:1431-6.
7. Miller RA, Masarie FE. The demise of the "Gre Oracle" model for medical diagnostic systems. Meth Infor Med 1990;29:1-2.
8. Hripcsak G. Ludemann P. Pryor TA. Wigertz O Clayton PD. Rationale for the Arden Syntax. Comp Biomed Res 1994; 27:291-324.
9. Hripcsak G. Writing Arden Syntax Medical Log Modules. Comput Biol Med 1994; 24(5):331-63.
10. Hripcsak G, Johnson SB, Clayton PD. Desperate seeking data: knowledge base-database links. In: Safran ed. Proc SCAMC 17, 1994; 639-43.
11. Pryor TA, Hripcsak G. Sharing MLM's: an experime between Columbia-Presbyterian and LDS Hospital. I Safran C, ed. Proc SCAMC 17, 1994; 639-43.
12. Cimino JJ. Johnson SB. Hripcsak G. Hill CL. Clayt PD. Managing vocabulary for a centralized clinical syste Medinfo. 8 Pt 1:117-20, 1995.
13. Cimino JJ. Clayton PD. Hripcsak G. Johnson Knowledge-based approaches to the maintenance of a lar controlled medical terminology. JAMIA 1994; 1(1):35-50
14. Humphreys BL. McCray AT. The Unified Medic Language System. Lindberg DA. Meth Inform M 1993;32(4):281-91.
15. Hripcsak G. Allen B. Cimino JJ. Lee R. Access to dat comparing AccessMed with Query by Review. JAMI 1996; 3(4):288-99.
16. Safran C, Porter D, Lightfoot J, et al. ClinQuery: system for online searching of data in a teaching hospit Ann Intern Med 1989; 111(9):751-6.
17. Pryor TA, Warner HR, Gardner RM, Clayton PD, Ha PJ. Software systems in HELP for development of hospit applications. In: Blum B, Orthner H, eds. Methods f Developing Clinical Information Systems. New Yor Springer-Verlag, 1987.
18. Timmers T, van Mulligen EM, van den Heuvel Integrating clinical databases in a medical workstatic using knowledge-based modeling. In: Lun KC, et al, ed Proc MEDINFO 92, 1992; 478-82.